NTNU

Kunnskap for en bedre verden

IDG2001 - Cloud Technologies

Sankini Rancha Godage

IDG2001 - Cloud Technologies

Course content

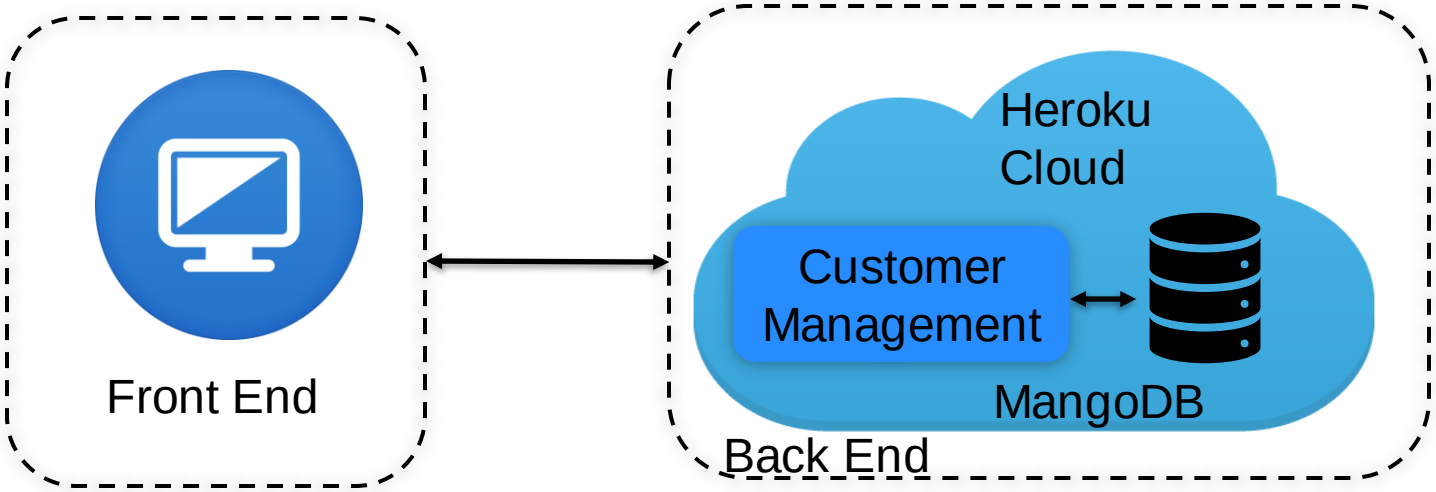
The field of cloud computing and cloud technologies is dynamic. It changes rapidly. There are some fundamental concepts that stay relatively unchanged, and there is an innovation in tools and technologies that often change. Due to those characteristics, the course lectures are subject to change, based on contemporary market adjustments. The current coverage includes, but is not limited to:

- networking in general, networking stack (OSI)
- **virtualization technologies**
- **basic use of Unix/Linux systems (e.g.shell, Bash, NFS, SSH)**
- **cloud and networking architectures**
- **Principles of Cloud Technologies (e.g. Infrastructure, Scaling)**
- Security
- **SaaS, PaaS, IaaS**
- **Economics Cloud Systems (Business models, Pricing models)**
- **cloud-based web APIs**
- In the practical sessions we plan to cover the course content.

IDG2001 - Cloud Technologies

- Why do we need VirtualBox?
- Why do we use Heroku?
- Why do we use mongoDB?

IDG2001 - Cloud Technologies



IDG2001 - Cloud Technologies

https://www.bacancytechnology.com/blog/all-about-full-stack-development

IDG2001 - Cloud Technologies

- Ajax

AJAX is a set of (typically) client-sided web development techniques, while REST is an architecture style for sending and handling HTTP requests. So you can use AJAX to send RESTful requests. A REST API is typically not implemented using AJAX, but can be accessed by an AJAX client.

IDG2001 - Cloud Technologies

Let’s see previous practical sessions

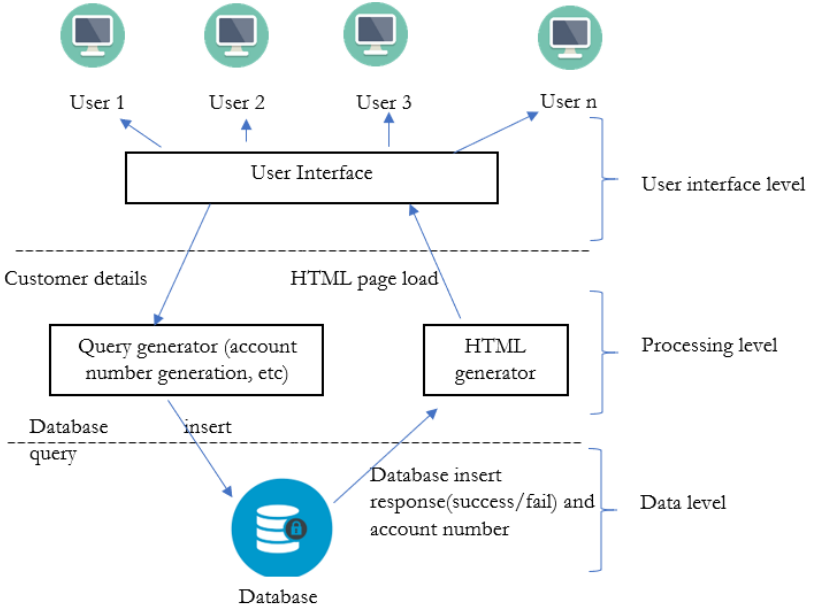
IDG2001 - Cloud Technologies

Project

- The bank called ‘ABC Bank’ is a large international banking firm which has multiple branchers in different cities. It offers many financial services including general banking, online banking, investments, credit cards and etc. ABC bank wants to expand their business and move to cloud computing technology.

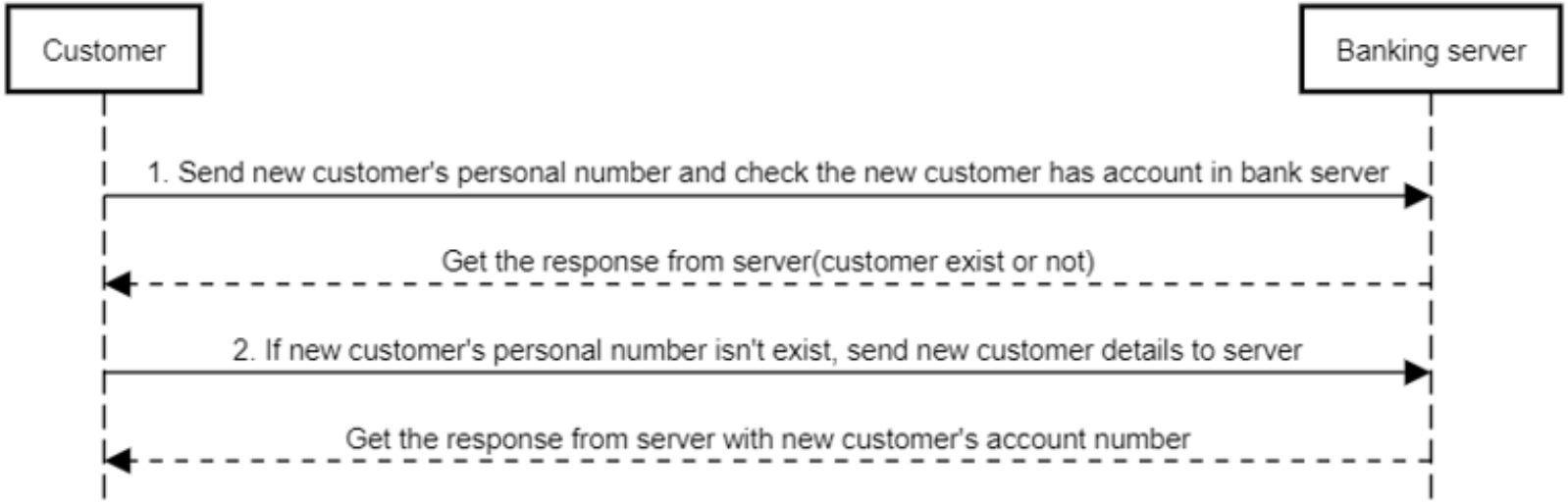
- The scenario: when the new customer wants to open the new bank account.

Sample client-server architectural



IDG2001 - Cloud Technologies

- Sample sequence diagram for banking system



IDG2001 - Cloud Technologies

Tasks,

- *Clome the sample project:
  git clone https://sankinir@bitbucket.org/sankinir/cloud.git
- You need to connect mongoDB to your project.
  https://www.mongodb.com/cloud/atlas
- Create database according to your requirements. (personal_number, first_name, last_name, data_of_birth,city,account_number ,created_date)
- Check the customer is exists or not (using customer personal number).
- If customer isn't exits in DB, insert customer details.
- Display the customer account number.
- Update the customer details(example, customer last name).
- Delete the existing customer using customer personal number.
- Deploy the project to Heroku server.

IDG2001 - Cloud Technologies

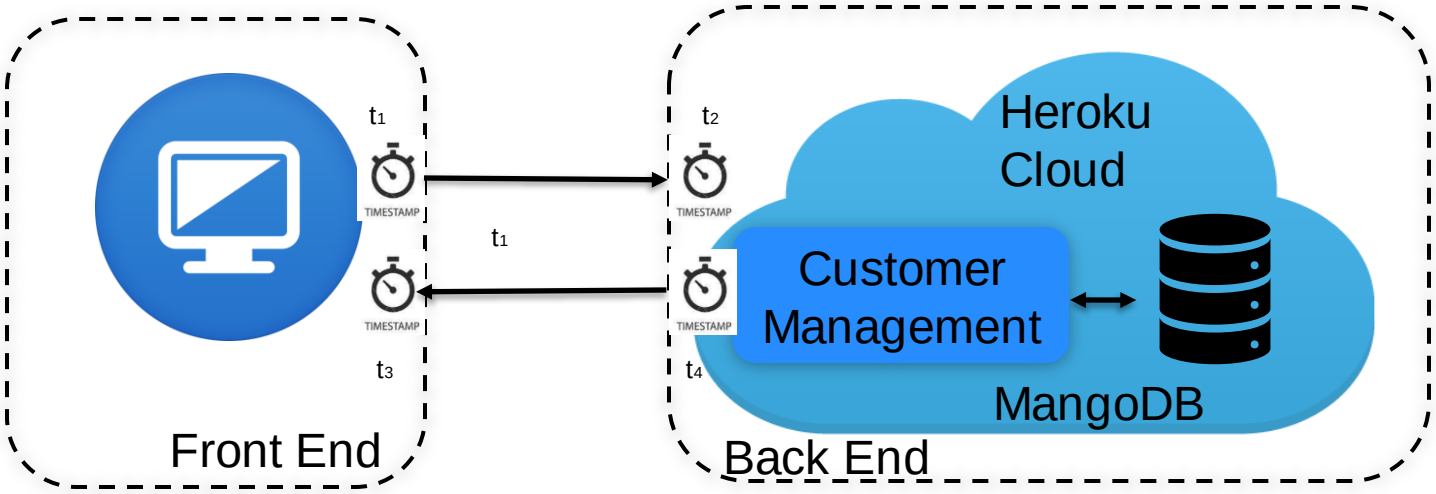
*Clome the sample project:

- npm install express
- npm install ejs
- npm install ejs
- npm install mongodb --save

IDG2001 - Cloud Technologies

Additional points (5 points):

Measuring latency breakdown for your running system.



IDG2001 - Cloud Technologies

Additional points (5 points):

Measuring latency breakdown for your running system.

End-to-end latency = t_3-t_1

Cloud processing latency = t_4-t_2

Communication latency = (t_3-t_1) - (t_4-t_2)

<payload size can be considered>

IDG2001 - Cloud Technologies

Suggested Schedule:

Week 2- Development of a basic running system

Week 3 - Preliminary evaluation,

Week 4 - Final deliverables

IDG2001 - Cloud Technologies

Project submission:

1. You have one month to complete the working project.
2. Only one group member should upload the project.
3. Report is not mandatory; You can upload project with Heroku URL.
4. Project submission deadline: 24th March 2021. **(PLEASE RESPECT DEADLINES)**

IDG2001 - Cloud Technologies

Marks

- Client server design and validation – 5 points
- Database design – 2 points
- Database connection – 2 points
- Client-server connection – 2 points
- At least 1 REST API(https://restfulapi.net/) should use(get, post, put, delete) - 7 points
- At least 1 DB query should work – 5 points
- Deployed to Heroku server – 2 points
- Additional task – 5 points